MYDOS 3.0 User Guide

Revision 3.013

Charles Marslett

WORDMARK Systems
2705 Pinewood Dr.
Garland, TX 75042

August 8, 1984

CONTENTS

MYDOS 3.0 User Guide

Revision 3.013

Charles Marslett

I. INTRODUCTION

The disk operating system described in this manual is for the ATARI (Trademark of Warner Communications, Inc.) 400 800, and XL series home computers. It is modeled after the Atari disk operating systems (DOS and DOS II) and may be considered an extension of the very "user friendly" concepts introduced with those two operating systems. For this reason the system documented here is identified as MYDOS 3.0. The ATARI 810 disk drive is well supported by the DOS II operating system, but DOS II has only limited provision for a double density disk system (the ATARI 815 dual drive system), and no provision at all for a system that could change density dynamically (a "dual" density system). Since dual density systems have now been manufactured for some time for the ATARI, and there does not appear to be an easy to use system to take full advantage of such a drive. WORDMARK Systems has developed a disk operating system to allow for dynamic density changes and larger capacity diskettes even in a single drive configuration.

II. SYSTEM REQUIREMENTS

MYDOS 3.0 requires 16K of RAM (provided by most 400, and all 800 or XL series computers). It also requires one ATARI 810 compatible disk drive or disk controller, and if the dynamic density selection is used, the controller must provide the extended 810 interface described in Section 10. This extended interface is supported by the PERCOM dual density disk subsystems, the ATR8000 disk/printer/RS232 controller manufactured by SWP, Inc. We know of no other disk drives or controllers that fully implement the necessary functions though.

Up to eight disk drives may be accessed, but only one is required. The resident part of the operating system leaves more free memory than the ATARI DOS or DOS II do. MYDOS supports all documented functions of the DOS II operating system, so MYDOS 3.0 supports most available software for the ATARI home computers. The BASIC and Assembler/Editor Cartridges both take advantage of the added memory available, and APX Pascal, the Deep Blue C Compiler, OSS BASIC A+, MEDIT and the AMAC macro assembler all run from double density disks with no alterations or limitations. ATARI BASIC has over 32,400 bytes left in a 40K system, even with eight double density disk drives configured.

The memory occupied by the permanently resident part of MYDOS is $0700 to $1E00 and that part occupied by the utility program exchanged with the MEM.SAV file extends to $3500. Unlike ATARI DOS, MEM.SAV need not be used if a program is to run directly from the disk (or cassette) and contains code located between $1E00 and $3500: the 'L' command will overwrite that part of memory. However, if the program is to be patched or run manually with the 'M' command, it must be loaded into MEM.SAV using the 'N' command because the menu program will be loaded back into memory on top of it before it can be modified or used!

The memory available to a program is affected only by the number of files to be open concurrently: each disk file that may be open at the same time requires 256 bytes of buffer space. Note that memory requirements are independent of the number of disk drives or the sector size (density), however.

III. MENU FUNCTIONS

The menu provided by MYDOS 3.0 identifies 18 common tasks that might need to be done. Rather than having to write a utility program (only a few lines of BASIC would perform most of the menu functions) or even remember the name and format of a DOS command, these tasks can be handled by entering a single letter. MYDOS 3.0 responds with a question asking for the details of the operation (which file, what density, "are you sure?" or whatever else it might need to know). After you enter the remaining information, the function is performed and another prompt is displayed.

You should notice two interesting things about the menu: the second line on the screen identifies the disk drives present on the system and what they appear to MYDOS 3.0 to be (single or double density). The next line describes the current default directory (that directory used when a disk is referenced by 'D:' (without a unit number after the 'D'). The second thing to notice is that after commands fill the screen, the menu 'rolls' off the top: some DOS programs keep the menu, MYDOS 3.0 does not. This permits more information to be shown on the screen when a long sequence of commands is needed to perform a function or when a directory is being listed or several files are being copied. To restore the menu to the screen, just type RETURN and the initial screen is restored.

Some commands require further information to prevent accidental damage to your disk files: the 'I' command and the 'J' command both require confirmation (through an additional key entry) before distroying the destination disk. To abort either operation without damaging any existing disk files, simply press RESET or BREAK. The 'D', 'E', 'F' and 'G' commands (delete, rename, lock and unlock) all require an explicit file specification. All other commands assume the drive containing the default directory or all the files in the default directory

(depending on whether the command affects an entire drive or a set of files).

Disk drive specifications and file specifications are made using the same rules: if only a drive is specified and file data is required, all files on that drive (or in the case of the 'K' 'L' and 'N' commands to save and load programs, the first file on that drive) will be the assumed choice. A drive is specified with a ':' (meaning the default drive), a number (with or without a trailing ':') or the capital letter 'D' followed by an optional number and a required ':'. If you wish to specify the file or set of files to be referenced, the drive format must include a ':' or it must be omitted entirely --

EXAMPLES: D1:Test.obj, 1:TEST.ASM, or D2TEST (really D1:D2TEST) are valid file names, but d1:Test.obj or 1TEST.ASM are not.

The file name itself is either fully specified (refering to exactly one file on the disk) or includes wild cards (specifying a set ofzero or more files). A fully specified file name consists of one to eight characters followed by a period ('.') and zero to three additional characters. The first character in the file name must be an upper or lower case letter, an underscore ('_') or the characters '@' or ''. The '' is the ATARI diamond graphic. The remaining character may be in that set or one of the digits 0-9. The 'wild card' characters are the characters '*' and '?': the character '*' or the sequence '*.' end either the 8 character or the 3 character field in the file name and match all possible characters. The character '?' matches any single file name character.

In addition to the main directory (containing up to 64 files or directories) each MYDOS 3.0 disk may also contain additional directories of 64 files each. If the main directory contained the directory file BAS and the file GRAPHIC1 were in the directory file BAS, it could be referenced with the filename, BAS:GRAPHIC1. If instead, GRAPHIC1 were in the directory GR.dir which in turn were in BAS, then the reference would be to BAS:GR.dir:GRAPHIC1 (and so on with as many names as needed). Because there is no limit to the number of directories on a disk (other than the buffer size of programs using the directories and number of available sectors on the disk), a single diskette can contain hundreds of files if necessary. Each directory is limited to 64 files or subdirectories though.

If a disk directory includes the files TEST.ASM, TEST.OBJ, TEST.C, TEST.ALM, TEASET.DOC, TRACE.FIL, and BETS.LST, the specification "t*.*" will not match any file name (since "t" and "T" are not the same letter to MYDOS 3.0). The specification "T**" will match all but "BETS.LST" (since the others all begin with the letter "T"). The specification "?E??.*" will match the first four files and the last one (since the 8 character part of the file name must have no more than 4 characters in it and the second character must be an "E"). The specification "*?" will match only the file TEST.C

(since it is the only file name with a single character in the 3 character field). The specification "????E**" will match the files TEASET.DOC and TRACE.FIL and none of the others (since the 8 character part of the file name must have at least 5 characters and the fifth must be an "E").

Where more than one file name is asked for, the first may be omitted by starting the response with a space or comma, and the last may be omitted by ending the line with a comma (the space cannot be used here since trailing spaces are ignored). If both file names are entered, they may be separated with either a space or a comma. Some commands may be modified using a letter following the character '/' after the file name (for example, 1/A or D1:TEST/A). The letter used (the modifier) generally means the same thing if it is allowed. Invalid modifiers are always ignored with no error indication at all. The modifier '/A' causes the results to be appended to the end of an existing file. This is applicable to 'C' (copy) and 'A' (directory) commands. The modifier '/N' causes the destination disk formatting to be skipped (saving about a minute) when used in the 'I' (initialize) and 'J' (duplicate disk) commands. It prevents the questions asked before changing each file if it is used in the 'D' (delete), 'F' (lock) and 'G' (unlock) commands; so we bend the rule only recently mentioned. In both cases, part of the function is skipped (see?). The '/X' command causes MYDOS 3.0 to pause at the end of each read or write pass when copying data to allow you to change disks (permitting you to copy from one disk to another with a single drive, even if that drive appears to MYDOS 3.0 to be two drives). This option supports disk drive that handle increased capacity by making a single diskette appear to be on more than one drive at a time, as well as disk drives that select density through the drive number (1-4 are single density, 5-8 are double, for example, and drives 1 and 5 are the same physically). This also allows the writing of a directory of one disk onto another as a file (use the command 'A' followed by the entry "1,1:DRV01.dir/X") even with only a single drive on the system. The '/X' is assumed if only one file name is entered in the copy command. This emulates the operation of the ATARI DOS II 'O' command which is not implemented in MYDOS 3.0.

To omit copying files with extensions beginning with 'S', the '/S' modifier can be appended to the source specification in the 'C' (copy) command: for example, the line "2/S,1" will copy all files not matching the string *.S?? from drive 2 to drive 1.

IV. THE MENU COMMANDS

A. List a Directory or a Set of Files

The 'A' command will list the files on a disk with their sizes, followed by a line specifying the number of free sectors on the

disk. If the line starts with an '*', the file has been locked and may not be modified or deleted without first being unlocked. A ':' before the file name marks those files that are directories. These files cannot be read or written as other files but only accessed as directories. No indication is made of the format of the file (ATARI DOS, ATARI DOS II, or MYDOS 3.0 are the three supported file formats). See Section 6 for further information if you need more information about the files than the 'A' command provides. This command will list the directory information to the screen if only one file specification is entered. If two are entered, the second is taken as a destination file and will be overwritten (or appended to) with the directory data: the entry "1,P:" will write the directory of the disk on drive 1 to the printer, for example.

To list the files in a subdirectory, enter the name of the directory followed by the symbol ':'. For example, "1:TEST:BAS:" will list the files in the directory BAS which in turn is in the directory TEST in the main diskette directory.

B. Run the Cartridge

The 'B' command returns control to the cartridge in the first cartridge slot. If no cartridge is present, an error is displayed, and nothing happens. No additional information is required, so if a cartridge is present it is entered after loading MEM.SAV (if the last load command was an 'N') or immediately (if the last load command was an 'L').

C. Copy a File or a Set of Files

The 'C' command is used to make another copy of one or more files of data. The two file specifications asked for after entering the 'C' identify the source and the destination of the information being copied. Either may be fully specified disk file or a device specification (such as E:, P: or one of the RS232 ports R1: to R4:). The destination may be a set of disk files (specified with '*' and '?'s) only if the source specifies a file name for the destination to use. Copies from a file set to a device will implicitly write consecutive files to the device (generating a set of listings or a collection of cassette files for example). The source may be a set and the destination a single disk file, but unless the '/A' modifier is specified to append each copied file to the end of the previously copied files, only the last source file will remain on the destination disk.

Note that the 'C' command always uses the full memory space for a copy operation (unlike ATARI DOS II) and as a result, will always invalidate MEM.SAV if it is used. Any pending program cannot be restarted after a 'C' or 'J' command.

D. Delete a File or a Set of Files

The 'D' command will remove all files that match the file specification entered asking for confirmation before each one is removed. This verification that the file is really the one to be removed can be disabled for the duration of this single 'D' command by adding the command modifier '/N' to the end of the file specification. In this case, all the matching files will be removed 'quietly' and the only further indication you will see is the prompt for the next command.

E. Rename a File or a Set of Files

The 'E' command changes the name of the source file or files to match the specification in the destination. Unlike other file specifications, the destination specification must consist of a single file name: it must not contain any directory names or a disk drive specification. For example, "D2:TEST:BASIC:NOTPNT.BAS,RANDIO.BAS" is the line entered to change the name of a file in the directory "D2:TEST:BASIC". To change the name of the directory "BASIC" to "ATBASIC", the line would look like "D2:TEST:BASIC,ATBASIC".

F. Lock a File or a Set of Files

The 'F' command limits access to the files identified. The files may not be deleted, renamed, added to or replaced without being first unlocked with the 'G' command. When a directory is listed, the files that have been locked using either the 'F' command or the 'lock' or 'open locked' functions provided through CIO will be marked with an '*' in the first column. The files that are locked may be read or loaded and executed normally, only modification or removal are prohibited.

Before each file is locked MYDOS asks you for confirmation with a message: for the file TEST, the message would read "Lock TEST?". Any answer but 'Y' will result in the file not being locked. The confirmation questions can be skipped by adding '/N' to the end of the file specification.

G. Unlock a File or a Set of Files

The 'G' command removes the limitations imposed on a file when it is 'locked' using the 'F' command. It does not alter the file or otherwise change the way the file is accessed or used. The same function may be performed in a program through the CIO function to 'unlock' a file.

Before unlocking each file MYDOS asks for confirmation with a question that must be answered with a 'Y' if the file is to be unlocked; otherwise, no action is taken and the next confirmation question is asked. To disable the confirmation

questions, enter '/N' after the file specification (see Section 4.4, on deleting files, for a more detailed explanation).

H. Write MYDOS 3.0 to a Disk

The 'H' command is used to make a rebootable copy of the current MYDOS 3.0 files in memory. The two files created or rewritten are 'DOS.SYS' and 'DUP.SYS'. 'DOS.SYS' is an image of the permanently resident file management routine accessed through CIO and the small interface package that loads and saves MEM.SAV (an image of the part of memory used to hold the nonresident part of MYDOS 3.0) and the second part of MYDOS 3.0 itself ('DUP.SYS'). The file 'DUP.SYS' is a standard load file containing the part of MYDOS 3.0 that is overwritten when a program is loaded into memory. Neither of these files is compatible with any other disk operating system either for the ATARI or any other home computer. Both should be treated as a single object. Never copy only DOS.SYS or only Dup.SYS to a disk without copying the other. The files written to the disk by the 'H' command will reflect the configuration parameters currently in memory, which may be different from the ones active if the system were rebooted from disk again. (See Section 9 for the definition of the configuration parameters provided in the system and how to specify a modified configuration.)

I. Initialize a Disk

The 'I' command is used to prepare a new disk for use with the MYDOS 3.0 operating system or to remove all the files on an old disk. The result of the 'I' command is a completely empty disk. The only data on the diskette is that system provided information defining the space available and the empty main directory. If the drive number is followed by a '/N' modifier, the diskette will not be reformatted, but just 'erased'. This is the recommended way to remove all the files on a diskette, rather than to use the 'D' command.

J. Duplicate a Disk

The 'J' command copies all the information from one diskette to another. The information on the diskette is identified by the MYDOS 3.0 bit map (on the VTOC sectors) if no sector range is specified. If a range is specified, the data to be copied are the sectors in that range and the VTOC is not examined. Specifying a sector range is done by adding two numbers seperated by a dash and enclosed in parentheses to the end of the drive specification(s). For example, to copy sectors 19 through 54 (tracks 1 through 3) from drive 1 to drive 3 the command line could be "1,3(19-54)".

The disk initialization done by the 'J' command is done without

error checks: this means that a disk formatted with the 'J' command may have bad sectors (in the case of creating a backup disk, the disk will not be written to later so this is acceptable). If the disk is to be a working disk, a more reliable approach is to initialize the disk (with the 'I' command) and then copy the data using the 'J' command and the '/N' modifier (see the next paragraph).

If the destination disk is already a properly formatted MYDOS 3.0 diskette, the '/N' modifier may be entered after either drive number to skip the formatting of the destination drive. Otherwise, the destination diskette will be formatted before the data from the source is copied to it. That is, either "1/N,2" or "1,2/N" will copy from drive 1 to drive 2 without first formatting the diskette in drive 2. To copy the first two tracks of a diskette without formatting the diskette being copied to, you could enter "1,2/N(1-36)" or "1 2/N(1-36)".

Note that the 'J' command, like that in ATARI DOS II, will use all of available memory to duplicate the diskette: this means that if memory has been saved using the MEM.SAV file, it will no longer be valid. Any pending program cannot be restarted after a 'C' or 'J' command.

K. Save Memory to Disk

The 'K' command builds a binary load file containing the data from the memory area specified, as well as an initialization and a run vector address if specified. If the file is not to execute an initialization routine on being loaded, the initialization vector should be omitted. If it is not to run on being loaded, the run vector should also be omitted (and trailing commas need not be typed in either). If either vector is entered as zero, that vector will not be invoked when the program is loaded. Note that the starting and ending addresses of the program and the initialization and run entry points of the program are all specified as hex numbers.

If MEM.SAV is active when the 'K' command is entered, the MEM.SAV file is loaded before writing the file to the disk.

L. Load Memory from a File

The 'L' command takes a binary load file from the disk and loads it into memory. The load file's initialization routine(s) will be executed and the program started at its run address unless the '/N' modifier is appended to the file name. This command disables the MEM.SAV file before loading and executing the program.

M. Run at an Address

The 'M' command is used to enter a program loaded without a run address, or to jump into any program without the need for a return address. It may be used to restart the computer (loading the AUTORUN.SYS file, if any) by specifying $E477 as the jump address. If MEM.SAV is active (enabled with the 'N' command and not since disabled by the 'L' command), the contents of memory will be restored from MEM.SAV before jumping to the address specified.

N. Load MEM.SAV from a File

The 'N' command takes a binary load file from the disk and loads it into memory. The load file's initialization routine(s) will be executed and the program started at its run address unless the '/N' modifier is appended to the file name. This command enables the MEM.SAV file before loading and executing the program and when control is returned to MYDOS the contents of memory will be saved in MEM.SAV.

If no file name is specified, the MEM.SAV file usage is enabled but no program is loaded or run.

O. System and Drive Configuration

The 'O' command is used to specify the type (at least logically) of the disk drives on the ATARI computer. Additionally, it is used to specify the number of file buffers provided and to control verification after disk write operations. For owners of early production 800 and 400 computers with the ATARI A revision OS ROMs this command is used to disable the 'fast write' algorithms that do not reliably work with those computers. Also, by disabling fast writes, code in ROMs may be written to disk (say to later be disassembled by any of several BASIC disassemblers). This is necessary since the fast write (or "Burst mode I/O" as referred to in ATARI and OSS documentation) operations modify the text buffer, then restore it, during a write and this is obviously not possible if the text is in ROM. These three functions, which are not specific to individual drives, are selected by entering a RETURN when the prompt asking for a drive number is displayed. Three questions will then be asked: "Verify WRITEs" expects a 'Y' to be entered if all data written to the disk is to be read back to verify that it was not only written correctly, but that the data is in fact readable from the disk. The second question: "Number of File Buffers" expects a number, followed by a RETURN to specify the number of file buffers to be allocated. The third question, 'Fast WRITEs allowed?', should be answered 'Y' if the computer is an XL series computer or a 400 or 800 computer with the B-revision OS ROMs. If the computer is an early 400 or 800 computer with the A-revision ROMs the answer should be 'N' to disable the fast write algorithm because the A-revision ROMs do not allow the sector buffer to be located anywhere in memory. As mentioned above,

answering 'N' will also permit SAVEing memory in the OS ROMS or in cartridges using the 'K' command. A RETURN does not retain the current value -- the result is YES as the answer to the 'Verify WRITEs' question, and 3 buffers as the answer to the number of buffers question. Anything but a 'Y' will enable fast writes in response to the third question.

If instead of a RETURN, a drive number or name had been specified then that drive would be reconfigured. The first question identifies whether the drive is to be included in system initialization (and thus be available for later use). If a non-existent disk is included it does not cause any problems with the system: it simply causes that disk to be examined each time the system is booted (adding perhaps a second to the time it takes to boot up MYDOS 3.0). If drive is excluded from the system, no further questions are asked. Otherwise, the second question asks if the drive is configurable: that is, is it like the ATARI 810 or 815 drives (with a fixed configuration) or is it like the PERCOM or ATR8000 drives. If the disk is not configurable it is assumed to be a 720 sector, single or double density ATARI 81x disk drive. Drives excluded from the system can be dynamically added by referencing them but they will always be treated as 5 1/4 inch 810 or 815 compatible drives (the default configuration).

If the first two answers are 'N' (do not exclude the drive) and 'Y' (it is configurable), the configuration is asked for: Is the drive double sided, how many tracks are there on the each side of the disk, and at what speed can it move the read/write head across the disk (what is its step rate). The first question is answered with 'Y' or 'N' ('Y' meaning 'yes' it is a double sided drive and diskettes formatted on it will be double sided). The second question is answered with 35, 40, or 80 followed by a RETURN if the disk drive is a 35 track, 40 track, or 80 track 5 1/4 inch floppy drive and with 77 if the drive is a standard 77 track 8 inch drive. No other number is accepted. The answer to this question specifies both the type of drive (8 inch or 5 1/4 inch) as well as the number of tracks per inch and total capacity of the drive. This answer is very important to the operation of the drive. The last answer is entered as a code: use the following table and the drive specifications to determine the proper value.

Code value	8 inch rate	5 1/4 inch rate
0	3 ms/track	6 ms/track
1	6 ms/track	12 ms/track
2	10 ms/track	20 ms/track
3	15 ms/track	30 ms/track

P. Disk Density Selection

The density used for most MYDOS commands is determined by the data written on the diskette and the operator need not worry about setting it. The 'P' command is provided to allow forcing

the density setting for the format ('I') command and to permit setting the density before loading a program that does not automatically select the proper density (most programs do not).

MYDOS commands that access a diskette will automatically select the appropriate density, so the 'P' command will have no effect on the drive if any command accessing the drive configured with the 'P' command is executed before the format ('I') command or the program is run (using the 'B', 'L' or 'N' commands).

Q. Create Additional Directories

When a diskette is formatted, an empty directory (the highest level or root directory) is created (empty). This directory is capable of holding up to 64 files or other directories. If additional directories are installed in this directory, each of the additional directories can contain up to 64 files as well. A directory is installed in an existing directory using the 'Q' command and responding to the question of what the directory name is with the name of the new directory. For example, if "TEST" and "BAS" are two directories in the root directory of the diskette in drive 1, "1:TEST:COMM" or "1:BAS:COMM" would both create a new directory in "TEST" and "BAS" respectively. "1:NEW:COMM" would not, however, since "NEW" does not already exist. A 'Q' command with the response "NEW" would create the first directory followed by an additional 'Q' command with the response "NEW:COMM" would create the two nested directories, though.

Each directory takes up 8 sectors and after it is created it may only be referenced as a directory (folowed by a ':' that is) or deleted. It may only be deleted if it is empty (if it has no files in it). A directory may be emptied by using the 'D' (delete) command and specifying the files "*.*/N" to remove all the files in the directory.

R. Set the Default Directory

The 'R' command is used to select a directory to be used when a file is referenced without the drive number: that is, when file names such as "TEST1.BAS" or "D:NEWCODE" or even ":BIGFILE" are used, they are assumed to be in the default directory. Programs run under MYDOS 3.0 can access the contents of the current default directory by using a file name of the form "D:..." without the drive number explicitly entered. They may also set the default directory by calling the CIO Function code 41 (set directory) routine.

The directory is set by inserting the diskette with that directory on it into the desired drive, then entering the file name of the directory with no trailing ':'.

If the diskette in the drive containing the default directory

is replaced, the default should be redefined unless it is the root directory. This is because only the root directory is at the same location on all diskettes.